



## AMD CodeXL 1.4 GA Release Notes

Thank you for using CodeXL. We appreciate any feedback you have! Please use the [CodeXL Forum](#) to provide your feedback.

You can also check out the Getting Started guide on the [CodeXL Web Page](#) and the latest CodeXL blog at [AMD Developer Central - Blogs](#)

This version contains:

- For Windows for 32-bit and 64-bit Windows platforms
  - CodeXL Standalone application
  - CodeXL Microsoft® Visual Studio® 2010 extension
  - CodeXL Microsoft® Visual Studio® 2012 extension
  - CodeXL Microsoft® Visual Studio® 2013 extension
  - CodeXL Remote Agent
- For 64-bit Linux platforms
  - CodeXL Standalone application
  - CodeXL Remote Agent

### Note about 32-bit Windows CodeXL 1.4 Upgrade Error

On 32-bit Windows platforms, upgrading from CodeXL 1.0 using the CodeXL 1.4 installer will remove the previous version and then display an error message without installing CodeXL 1.4.

The recommended method is to **uninstall previous CodeXL versions before installing CodeXL 1.4.**

If you ran the 1.4 installer to upgrade a previous installation and encountered the error mentioned above, **ignore the error** and run the installer again to install CodeXL 1.4.

### Note about installing CodeAnalyst after installing CodeXL for Windows

CodeXL can be safely installed on a Windows station where AMD CodeAnalyst is already installed. However, do not install CodeAnalyst on a Windows station already installed with CodeXL. **Uninstall CodeXL first, and then install CodeAnalyst.**

## System Requirements

CodeXL contains a host of development features with varying system requirements:

- **GPU Profiling and OpenCL Kernel Debugging**
  - An AMD GPU (Radeon HD 5000 series or newer, desktop or mobile version) or APU is required.

- The AMD Catalyst Driver must be installed, release 13.11 or later. Catalyst 13.35 is the recommended version. See "[Getting the latest Catalyst release](#)" section below.
- For **GPU API-Level Debugging**, a working OpenCL/OpenGL configuration is required (AMD or other).
- **CPU Profiling**
  - Time-Based Profiling can be performed on any x86 or AMD64 (x86-64) CPU/APU.
  - The Event-Based Profiling (EBP) and Instruction-Based Sampling (IBS) session types require an AMD CPU or APU processor.

Supported platforms:

- Windows platforms
  - Windows 7, 8 and 8.1, 32-bit and 64-bit
  - Note: For the CodeXL Visual Studio 2010/2012/2013 Package, the station must be installed with Visual Studio 2010/2012/2013, respectively. However, the CodeXL Standalone Application does not require Visual Studio to be installed.
- Linux platforms
  - Red Hat EL 6.4 64-bit
  - Ubuntu 12.04 64-bit

## Getting the latest Catalyst release

The way to get the latest beta driver is to use the links "Latest Windows Beta Driver" and "Latest Linux Beta Driver" on the Graphics Drivers support page:

<http://support.amd.com/us/gpudownload/Pages/index.aspx>

## New in this version

The following items are new in this version:

- Visual Studio 2013 CodeXL Extension
- New Shader Analyzer included in CodeXLAnalyzer command line tool (Windows only)
- Added support for Volcanic Islands family GPUs - debugging, profiling and analyzing.
- Analyze Mode
  - Streamlined Analyze Mode for quick creation and editing of CL files
  - New Build Options node in the Project Settings tree for easy control and visibility of offline OpenCL build settings
  - Analysis table displays separate clocks count for single Work Item and for the total workload
- User Experience improvements
  - New 'No Project Loaded' dialog for quickly creating a project and setting the relevant mode
  - Re-design for 'Project Settings' dialog
  - GPU Debugger's Locals view displays work group dimensions and properties during kernel debugging
  - GPU Profiler timeline modifications
    - User markers wrap API calls as expandable tree nodes

- New columns in the API Trace table
    - The Properties View shows the properties of the selected timeline item
    - Application Trace session type has been renamed to Application Timeline Trace
  - Option to delete multiple profiler sessions
  - CL files Drag n' Drop automatically adds the file to the CodeXL project
- CPU Profiling
  - Added support for Kaveri and Mullins APUs
  - New capability: Attach To Process
  - New capability: System Wide profiling without tying the session to a specific process.
  - Call graph view – simplified and corrected the Parents and Children table
  - Automatic fetching of disassembly blocks when scrolling through CPU Profiling source code/disassembly view
  - Callstack collection now collects code executed in kernel-mode in addition to user-mode code that was collected by previous CodeXL releases
  - Display indication of symbols loaded per module
- CLPerfMarker library replaced by a new generic logging API library AMDTActivityLogger
- OpenCL is no longer a prerequisite for doing OpenGL debugging

## Fixed Issues

The following fixes were not part of the 1.3 release and are new to this version:

- `__read_only` and `__write_only` Image variables are not shown in locals during GPU Debugging. (345852)
- Segmentation Fault while launching CodeXL in root mode on Llano (A-3000 series APU) + Radeon HD 6870. Fixed in Catalyst OpenCL driver 14.10.  
[Workaround for previous drivers: Rename the file /usr/lib64/libamdocl64.so.](#) (401418)
- CodeXL crashes while debugging `clBolt.test.StableSort` sample. Fixed in Catalyst OpenCL driver 13.35. (406071)
- CodeXL crashes when building APP SDK sample Concurrent Kernel in Analyze mode. Fixed in Catalyst OpenCL driver 14.10. (420158)
- On Catalyst driver 13.11 Beta and later, APP SDK sample `AESDecryptDecrypt` crashes when debugged. Fixed in Catalyst OpenCL driver 13.251. (420499)
- CodeXL Visual Studio 2010 extension crashes when “show disassembly” is checked in VS disassembly window and step-in is performed after breakpoint at `enqueueNDRangekernel` (423981)
- CodeXL crashes on running a system-wide ‘assess performance’ CPU profiling session with “ScimarkStable” sample and call-stack collection is enabled. (424061)
- CodeXL crashes on selecting any one core in CPU Profiling ‘Display Filter’ for a Cache Line Utilization profile session. (424145)
- CodeXL crashes after when breaking on APP SDK 2.8.1 sample `MatrixMulDouble` source code line 139. (423013)
- CodeXL Visual Studio extension crashes after step-in with breakpoint at `enqueueNDRangekernel` and log level set to Extensive. (424063)
- Remote GPU Profiling generates a redundant `atp` file in the session directory. (422859)
- On Linux, a debugged application’s stdout content is not displayed when the application is run by CodeXL debugger. (359816)

- When breaking a debug session, cannot view FBO objects such as Texture and Render buffer. (378010)
- "Stop Debugging" and "Break" commands are disabled while running Teapot sample for the first time. (387043)
- On Red-Hat Linux 64-bit Intel platforms, for CPU Profiling sessions with call-stack collection enabled, a "No Samples available" error message is shown while post-processing. (387068)
- Multiple paths separated by a ";" in Environmental Variables field of "Project Settings" are lost once a debug/profile session begins. (390788)
- During CPU Profiling on Linux, application execution halts if keyboard input provided using the console terminal. (360479)
- CodeXL 1.3 installer does not delete extracted files in C:\Users\username\AppData\Roaming\Advanced Micro Devices, Inc\AMD CodeXL 1.\*.\*\install after installation is complete. (416302)
- On Linux, the GPU Profiler displays zero for all Performance Counters for APP SDK samples BoxFilterGL and Mandelbrot. Fixed in Catalyst OpenCL driver 14.10. (407766)
- In CPU Profiling session, a functions appears in the "5 Hottest Functions" table, but not in the Call Graph's functions table. (401484)
- GPU Debugger step-in order is out of sequence with source code lines order in APP SDK 2.8.1 NBody sample. Fixed in Catalyst OpenCL driver 13.251. (404041)
- In a GPU Profiler Performance Counters session, mismatching ISA/IL shown for kernels of application containing multiple kernels. Fixed in Catalyst OpenCL driver 13.251. (405430)
- CPU Profiling results for Timer samples vary when comparing Time-Based-Profiling session to Custom session containing Timer and Event Based profiling together. (420887)
- In GPU Debugger's buffer view, the Hovered Value and Selected Value fields are active only for the lower left quarter of the image and displays the wrong value. (420968)
- For CPU Profiling Assess Performance sessions on Windows 8.1, occasionally an empty session is generated. (421115)
- CodeXL may fail to import .csv file generated from sprofile run on Linux machine. (422593)
- When CPU Profiling on an Intel Linux platform with Call-stack collection enabled, an error message appears: "Cannot write profile data file". (423985)
- On Linux KDE platforms, when GPU profiling the TeaPot sample on a remote target, the Kernel Occupancy view may fail to open. (424024)
- CodeXL takes a very long time to display Call-Graph view in a CPU Profiling Assess Performance session of the Teapot sample. (424036)
- In CodeXL Analyze Mode, clicking a node on the explorer tree does not focus on the right tab if it was already opened, and tabs fail to close when closing it. (422991)
- Load Selection In Performance Counters tab does not update the list of selected counters. (423054)
- Exiting CodeXL while GPU profiling is in progress doesn't close the profiled application. (423059)
- CodeXL GPU Debugger takes 5-10 seconds to step over a line of source code in a kernel file. (423061)
- CPU Profiling Source view does not reflect user selected function for Windows Store Apps. (423490)
- All views should implement copy/cut/paste operations and they should be accessible from the Edit menu. (420337)
- Analyze Mode fails to enumerate kernels in CL files that contain comments inside a kernel function definition or use void as a return type. (423684)

- CodeXL CPU Profiling Functions view shows huge invalid percentage bars. (401637)
- On Linux, for CPU Profiling Instruction-Based-Sampling sessions, some filters are missing from the drop-list in the 'Display Filter' dialog. (355074)
- If the focus shifts from the CodeXL VS Extension's "GPU Debug" options dialog to Visual Studio, the dialog is stuck. (362707)

## Known Issues

- Debugging OpenCL kernels that use read-modify-write atomic operations is not supported.
- GPU Debugging on OpenCL Static C++ Kernels is not supported. (334415)
- OpenCL 1.2 keyword printf and barriers are not supported during kernel debugging.
- Building kernels with OpenCL 1.2 clCreateProgramWithBinaries and clLinkProgram API prevents the display of source code when debugging these kernels. (369171)
- Performing CPU Profiling with Call-Stack Sampling (CSS) enabled, on systems with discrete graphics card (Radeon HD 5000, 6000 or 7000 series) and Linux kernel version 3.0 or lower, may result in Linux kernel panic. This kernel panic does not occur with Linux kernel version 3.2 onwards. (352399)
- CPU Profiling is disabled on Windows 8 and 8.1 if Hyper-V is enabled. (438549)
  - Note that installing Microsoft Windows Phone 8.0 SDK activates Hyper-V.
- CPU Profiling on Linux platforms - Limitations of PERF
  - CPU profiling uses PERF which requires kernel 2.6.32 or later. CPU Profiling with Call Stack Sampling requires Linux kernel 3.0 or later. However, we recommend using kernel 3.2 and above which has shown to be more stable.
  - Call chain analysis on Linux currently depends on the call chain information provided by Linux PERF. This requires the profiled binaries to have stack frame pointer. (i.e. compiled with -fno-omit-frame-pointer).
  - For non-root users to run CodeXL CPU profiling, `"/proc/sys/kernel/perf_event_paranoid"` needs to be set to `"-1"`.
  - Instruction-Based Profiling on Linux requires Linux kernel 3.5 and above, and must be used with system-wide profiling.
  - Call chain information (stack trace) for inline functions is not available.
- PERF call chains which contain call stacks across modules have shown to be truncated. This results in inaccurate "Deep Samples", "Downstream Samples", and "Call Path" analysis.
- If gDEDebugger 6.x is installed on the machine, mouse click doesn't start text fields editing in CodeXL Visual Studio Extension. [Workaround: Navigate to the text fields using TAB or uninstall gDEDebugger before installing CodeXL.](#) (344811)
- Menu items are present but not visible after minimization and restore of CodeXL in Ubuntu system using Unity theme. [Workaround: Use Unity 2D theme instead of Unity theme.](#) (353082)
- AMDTTeapot sample may crash while debugging OpenCL kernels after multiple step operations (45 or more). (357741)
- For applications that perform simultaneous asynchronous data transfer and kernel execution, the timeline shown in the Application Trace session view will not show these operations being overlapped. This is because the driver and hardware force these operations to be synchronous while profiling. (333981)

- CPU Profiling on Windows 8 shows two target applications in Profile Overview. The conhost.exe process is an actual executable. This process fixes a fundamental problem in the way previous versions of Windows handled console windows, which broke drag & drop in Vista.
- If CodeXL is installed in path that includes non-ASCII Unicode characters, profiling does not work (365118).
- GPU Profiler Performance Counters show wrong values for DirectX samples on Intel Platform. (378477)
- On Linux platforms, CodeXL crashes when an OpenGL application's window is closed during kernel debugging and a step operation is later performed (363957)
- APP SDK sample Concurrent Kernel crashes when debugged. (423041)
- Certain performance counters of the GPU Profiler may display inconsistent values when profiling on a Southern Islands GPU (Tahiti, Pitcairn, Capeverde, Oland). (400345)
- The GPU Debugger can't step into a kernel if blocks that contain a return statement. (422699)
- The kernel execution time reported by the GPU Profiler may not be reliable when running on an APU with active power management. (413379)
- During remote GPU Debugging, kernel source breakpoints from previous debug sessions may appear as active, but will not be hit.  
[Workaround: Before starting a remote GPU Debugging session, close all active kernel source windows and delete all existing kernel source breakpoints.](#) (422715)
- CodeXL goes to non-responding state while building an OpenCL file with many functions in Analyze Mode. (435067)
- CPU Profiling Call-Graph view and Source Code view may show different function names for the same function. (435839)
- Passing quoted command line arguments (with spaces) to the app being profiled, the app doesn't see them as a single argument. (437880)
- GPU Debugger does not break when breakpoint is set at barrier()/mem\_fence function. (432906)
- CPU Profiling multiple processes with call stack collection may result in call graph view displaying addresses instead of function names for functions used by more than one process. (441541)
- CodeXL Analyze Mode - the offline compiler does not free all resources allocated for a build. For users working with CodeXL Analyze Mode on large kernel development this means that every few builds they must shutdown CodeXL and rerun it to avoid a crash (out of memory) and continue working. (421550)
- A GPU Debugging session may crash when going into kernel debugging if OpenCL Runtime driver 14.10 (future release) is installed. (438036)
- A GPU Debugging session may not break on breakpoints located on lines that contain barrier(CLK\_LOCAL\_MEM\_FENCE)/ mem\_fence(CLK\_LOCAL\_MEM\_FENCE)/ mem\_fence(CLK\_GLOBAL\_MEM\_FENCE).
- GPU Debugging session crashes while debugging the APP SDK 2.9 Concurrent Kernel sample. (423041)
- CodeXL GPU Profiling Performance Counters session may cause system crash (Blue Screen - TDR) while profiling Firaxis's Civ6 Terrain demo. (412217)

## Support

Please use our [CodeXL Forum](#) for support requests and our development team will contact you.